PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellants:	Arjun Chandrasekar Iyer; Chandrakant Ramkrishna Bhavsar		
Assignee:	Siebel Systems, Inc.		
Title:	JOINLESS SQL QUERY OPERATION		
Application No.:	10/750,703	Filing Date:	January 2, 2004
Examiner:	Eliyah Stone Harper	Group Art Unit:	2166
Docket No.:	SBL0011C1US	Confirmation No.:	3820

Austin, Texas
Filed: June 30, 2011

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

This brief is submitted in support of the Notice of Appeal filed on May 18, 2011 by Appellants to the Board of Patent Appeals and Interferences from the Examiner's final rejection of claims 116-163. Appellants note that the Notice of Appeal filed May 18, 2011, was received by the USPTO, thereby giving appellants a period for filing set to expire on July 18, 2011.

Please charge deposit account No. 502306 for the fee of $540.00 associated with this Appeal Brief. Please charge this deposit account for any additional sums which may be required to be paid as part of this appeal.

REAL PARTY IN INTEREST

The real party in interest on this appeal is Oracle Corporation by virtue of acquisition the assignee, Siebel Systems, Inc.

RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences related to this application.

STATUS OF CLAIMS

Claims 116-163 are pending in the application.

Claims 1-115 were previously cancelled.

Claims 116-163 stand rejected in the Final Office Action dated January 19, 2011 ("FOA") and in the Advisory Action dated April 11, 2011 (the "Advisory Action").

Appellants appeal the final rejection of claims 116-163.

STATUS OF AMENDMENTS

Appellants submitted amendments on March 21, 2011, subsequent to the final rejection of January 19, 2011. As indicated in the Advisory Action, these amendments were not entered.

SUMMARY OF CLAIMED SUBJECT MATTER

The following summary of the claims is presented in accordance with 37 C.F.R. § 41.37(c)(v).

Appellants' independent claim 117 is directed to a method that receives at least one SQL statement, at a computer system. *See, e.g.*, page 7, lines 22-26 "called by the calling program." The at least one SQL statement is configured to operate on a first table and a second table. *See, e.g.*, page 18, lines 6-13 "obtaining input data;" step 710 of Fig. 7, page 11, lines 20-25 "input data;" and tables of Figs. 4 and 5. The at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table. *See, e.g.*, query method 622 of Fig. 6; element 730 of Fig. 7; page 19, lines 15-17 "join results."

The method also automatically generates a set of SQL statements, using a processor of the computer system, to query the first table and the second table. *See, e.g.*, page 19, lines 13-16 "In … step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "multiple SQL statements … can be created and executed," page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor." The set of SQL statements are based, at least in part, upon the at least one SQL statement. *See, e.g.*, page 19, lines 13-16 "In … step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed." The first table and the second table are stored in a computer-readable storage medium of the computer system. *See, e.g.*, page 7, lines 1-22 "data 108… stored in memory;" page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," etc. The automatically generating uses a relationship between the first table and the second table to generate the set of SQL statements. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented." The set of SQL statements comprises SQL statements

other than the at least one SQL statement. *See, e.g.*, page 23, lines 6-26 "multiple SQL statements ... can be created and executed," "new SQL statement is started for each SQL integration," etc.

The method also produces a first result set by querying the first table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying the first table is performed using the processor. *See, e.g.*, page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor." A second result set is produced by querying the second table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying the second table is performed using the processor. *See, e.g.*, page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor." The querying the first table and the querying the second table are performed without joining the first table and the second table. *See, e.g.*, page 23, lines 13-18 "multiple SQL statements without joins can be created and executed."

The method also joins the first result set and the second result set, using the processor, to produce a third result set. *See, e.g.*, page 19, lines 16-17 "Join Result Sets step 730 joins the results of the execution of each SQL statement to produce output data 109;" page 23, lines 13-18 "multiple SQL statements without joins can be created and executed, and the result sets joined;" page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor."

The method also returns the third result set in response to the receiving the at least one SQL statement. *See, e.g.*, page 19, lines 17-20 "In Provide Output Data step 740, output data 109 is provided. In one embodiment, output data 109 is provided in the form of SQL integration object instances."

Appellants' independent claim 128 is directed to a system that comprises a processor. *See, e.g.*, page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor." The system also includes a memory unit coupled to the processor. *See, e.g.*, page 7, lines 1-22 "data 108… stored in memory;" page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," Fig. 20.

The system also includes receiving means for receiving at least one SQL statement. *See, e.g.*, page 7, lines 22-26 "called by the calling program." The at least one SQL statement is configured to operate on a first table and a second table. *See, e.g.*, page 18, lines 6-13 "obtaining input data;" step 710 of Fig. 7, page 11, lines 20-25 "input data;" and tables of Figs. 4 and 5. The at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table. *See, e.g.*, query method 622 of Fig. 6; element 730 of Fig. 7; page 19, lines 15-17 "join results."

The system also includes generating means for automatically generating a set of SQL statements to query the first table and the second table. *See, e.g.*, page 19, lines 13-16 "In … step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "multiple SQL statements … can be created and executed." The set of SQL statements are based, at least in part, upon the at least one SQL statement. *See, e.g.*, page 19, lines 13-16 "In … step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed."

The generating means uses a relationship between the first table and the second table to generate the set of SQL statements. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented." The set of SQL statements comprise SQL statements other than the at least one SQL statement. *See, e.g.*, page 23, lines 6-26 "multiple SQL statements … can be created and executed," "new SQL statement is started for each SQL integration," etc.

The system also includes determining means for determining if a parent/child relationship exists between the first and second tables. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented."

The system also includes first producing means for producing a first result set by querying the first table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The system also includes second producing means for producing a second result set by querying the second table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying the first table and the querying the second table are performed without joining the first table and the second table. *See, e.g.*, page 23, lines 13-18 "multiple SQL statements without joins can be created and executed."

The system also includes joining means for joining the first result set and the second result set to produce a third result set. *See, e.g.*, page 19, lines 16-17 "Join Result Sets step 730 joins the results of the execution of each SQL statement to produce output data 109;" page 23, lines 13-18 "multiple SQL statements without joins can be created and executed, and the result sets joined."

The generating means, the determining means, the first querying means, the second querying means and the joining means reside in the memory unit. *See, e.g.*, page 46, lines 16-21 "system having... associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," etc.

The system also includes returning means for returning the third result set, in response to receiving the at least one SQL statement. *See, e.g.*, page 19, lines 17-20 "In Provide Output Data step 740, output data 109 is provided. In one embodiment, output data 109 is provided in the form of SQL integration object instances."

Appellants' independent claim 137 is directed to a computer program product that comprises receiving instructions to receive at least one SQL statement. *See, e.g.*, page 7, lines 22-26 "called by the calling program." The at least one SQL statement is configured to operate on a first table and a second table. *See, e.g.*, page 18, lines 6-13 "obtaining input data;" step 710 of Fig. 7, page 11, lines 20-25 "input data;" and tables of Figs. 4 and 5. The at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table. *See, e.g.*, query method 622 of Fig. 6; element 730 of Fig. 7; page 19, lines 15-17 "join results."

The computer program product further comprises generating instructions to automatically generate a set of SQL statements to query the first table and the second table. *See, e.g.*, page 19, lines 13-16 "In ... step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "multiple SQL statements ... can be created and executed." The set of SQL statements are based, at least in part, upon the at least one SQL statement. *See, e.g.*, page 19, lines 13-16 "In ... step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed."

The generating instructions are configured to use a relationship between the first table and the second table. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented." The set of SQL statements comprises SQL statements other than the at least one SQL statement. *See, e.g.*, page 23, lines 6-26

"multiple SQL statements ... can be created and executed," "new SQL statement is started for each SQL integration," etc.

The computer program product further comprises first producing instructions to produce a first result set by querying the first table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The computer program product further comprises second producing instructions to produce a second result set by querying the second table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying the first table and the querying second table are performed without joining the first table and the second table. *See, e.g.*, page 23, lines 13-18 "multiple SQL statements without joins can be created and executed."

The computer program product further comprises joining instructions to join the first result set and the second result set to produce a third result set. *See, e.g.*, page 19, lines 16-17 "Join Result Sets step 730 joins the results of the execution of each SQL statement to produce output data 109;" page 23, lines 13-18 "multiple SQL statements without joins can be created and executed, and the result sets joined."

The computer program product further comprises returning instructions to return the third result set, in response to receiving the at least one SQL statement. *See, e.g.*, page 19, lines 17-20 "In Provide Output Data step 740, output data 109 is provided. In one embodiment, output data 109 is provided in the form of SQL integration object instances."

The computer program product comprises a computer-readable storage medium. *See, e.g.*, page 47, lines 17-20 "Applications resident with computer system 10 are generally stored on and accessed via a computer readable medium, such as a hard disk

drive (e.g., fixed disk 44), an optical drive (e.g., CD-ROM drive 40), floppy disk unit 36 or other storage medium;" page 48, lines 8-10 "Code to implement the present invention may be stored in computer-readable storage media such as one or more of system memory 16, fixed disk 44, CD-ROM 42, or floppy disk 38."

The computer program product is encoded in the computer-readable storage media. *See, e.g.*, page 48, lines 8-10 "Code to implement the present invention may be stored in computer-readable storage media such as one or more of system memory 16, fixed disk 44, CD-ROM 42, or floppy disk 38."

Appellants' independent claim 146 is directed to a computer system comprising a processor to execute instructions. *See, e.g.*, page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor."

The computer system also includes a memory to store the instructions. *See, e.g.*, page 7, lines 1-22 "data 108… stored in memory;" page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," Fig. 20.

The memory is coupled to the processor. *See, e.g.*, page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," "bus," Fig. 20.

The instructions comprise receiving instructions to receive at least one SQL statement, at a computer system. *See, e.g.*, page 7, lines 22-26 "called by the calling program." The at least one SQL statement is configured to operate on a first table and a second table. *See, e.g.*, page 18, lines 6-13 "obtaining input data;" step 710 of Fig. 7, page 11, lines 20-25 "input data;" and tables of Figs. 4 and 5. The at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table. *See, e.g.*, query method 622 of Fig. 6; element 730 of Fig. 7; page 19, lines 15-17 "join results."

The instructions comprise generating instructions to automatically generate a set of SQL statements to query the first table and the second table. *See, e.g.*, page 19, lines 13-16 "In ... step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "multiple SQL statements ... can be created and executed." The set of SQL statements are based, at least in part, upon the at least one SQL statement. *See, e.g.*, page 19, lines 13-16 "In ... step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed."

The generated instructions use a relationship between the first table and the second table to generate the set of SQL statements. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented." The set of SQL statements comprises SQL statements other than the at least one SQL statement. *See, e.g.*, page 23, lines 6-26 "multiple SQL statements ... can be created and executed," "new SQL statement is started for each SQL integration," etc.

The instructions include first producing instructions to produce a first result set by querying the first table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The instructions include second producing instructions to produce a second result set by querying the second table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying instructions to the first table and the querying instructions to the second table are performed without joining the first table and the

second table. *See, e.g.*, page 23, lines 13-18 "multiple SQL statements without joins can be created and executed."

The instructions include joining instructions to join the first result set and the second result set to produce a third result set. *See, e.g.*, page 19, lines 16-17 "Join Result Sets step 730 joins the results of the execution of each SQL statement to produce output data 109;" page 23, lines 13-18 "multiple SQL statements without joins can be created and executed, and the result sets joined."

The instructions include returning instructions to return the third result set, in response to receiving the at least one SQL statement. *See, e.g.*, page 19, lines 17-20 "In Provide Output Data step 740, output data 109 is provided. In one embodiment, output data 109 is provided in the form of SQL integration object instances."

Appellants' independent claim 155 is directed to a computer system comprising a processor. *See, e.g.*, page 2, line 25 – page 3, line 8 "processor;" page 46, line 16 – page 47, line 23 "processor." The computer system also includes a memory unit coupled to the processor. *See, e.g.*, page 7, lines 1-22 "data 108… stored in memory;" page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," Fig. 20.

The computer system also includes a receiving module configured to receive at least one SQL statement. *See, e.g.*, page 7, lines 22-26 "called by the calling program." The at least one SQL statement is configured to operate on a first table and a second table. *See, e.g.*, page 18, lines 6-13 "obtaining input data;" step 710 of Fig. 7, page 11, lines 20-25 "input data;" and tables of Figs. 4 and 5. The at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table. *See, e.g.*, query method 622 of Fig. 6; element 730 of Fig. 7; page 19, lines 15-17 "join results."

The computer system also includes a generating module configured to automatically generate a set of SQL statements to query the first table and the second table. *See, e.g.*, page 19, lines 13-16 "In … step 720, SQL statements to retrieve data

corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "multiple SQL statements ... can be created and executed." The set of SQL statements are based, at least in part, upon the at least one SQL statement. *See, e.g.*, page 19, lines 13-16 "In ... step 720, SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed."

The generating module uses a relationship between the first table and the second table. *See, e.g.*, page 2, lines 8-11 "SQL adapter business service takes advantage of parent/child relationships between tables to construct SQL statements;" page 15, lines 9-22, various examples of "relationships;" page 23, lines 10-18 "SQL statement is generated for each table, producing simpler code and allowing a variety of relationships to be represented." The set of SQL statements comprises SQL statements other than the at least one SQL statement. *See, e.g.*, page 23, lines 6-26 "multiple SQL statements ... can be created and executed," "new SQL statement is started for each SQL integration," etc.

The computer system also includes a first producing module configured to produce a first result set by querying the first table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The computer system also includes a second producing module configured to produce a second result set by querying the second table using the set of SQL statements. *See, e.g.*, page 19, lines 13-16 "SQL statements to retrieve data corresponding to the search specification for each component are constructed and executed;" page 23, lines 13-17 "Alternatively, multiple SQL statements without joins can be created and executed...second approach is preferred because a SQL statement is generated for each table." The querying of the first table and the querying of the second table are performed without joining the first table and the second table. *See, e.g.*, page 23, lines 13-18 "multiple SQL statements without joins can be created and executed."

The computer system also includes a joining module configured to join the first result set and the second result set to produce a third result set. *See, e.g.*, page 19, lines 16-17 "Join Result Sets step 730 joins the results of the execution of each SQL statement to produce output data 109;" page 23, lines 13-18 "multiple SQL statements without joins can be created and executed, and the result sets joined."

The generating module, the determining module, the first producing module, the second producing module and the joining module reside in the memory unit. *See, e.g.*, page 46, lines 16-21 "system having… associated volatile and non-volatile memory;" page 47, lines 12-23 "RAM," "main memory," computer-readable medium," etc.

The computer system also includes a return output data module configured to return the third result set, in response to receiving the at least one SQL statement. *See, e.g.*, page 19, lines 17-20 "In Provide Output Data step 740, output data 109 is provided. In one embodiment, output data 109 is provided in the form of SQL integration object instances."

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The rejection of claims 116-163 under 35 U.S.C. § 103(a) as purportedly being unpatentable over U.S. Patent No. 6,438,542 ("Koo") in view of U.S. Patent No. 6,523,028 ("DiDomizio") is to be reviewed on appeal.

ARGUMENT

I. THE REJECTION OF CLAIMS 116-163 UNDER 35 U.S.C. § 103(A) AS BEING UNPATENTABLE OVER KOO IN VIEW OF DIDOMIZIO SHOULD BE OVERTURNED

Claims 116-163 are pending in the application and stand rejected under 35 U.S.C § 103(a) as being unpatentable over Koo in view of DiDomizio. While not conceding that the cited references qualify as prior art, but instead to expedite prosecution, Appellants have chosen to respectfully disagree and demonstrate that the rejections of the Final Office Action dated January 19, 2011 ("FOA") are clear error because the cited passages of Koo and DiDomizio, as well as Koo and DiDomizio generally, whether taken alone or in any permissible combination, fail to disclose, teach, or even suggest the limitations of the independent claims.

A. Claims 116, 117, 128, 129, 137, 138, 146, 147, 155, and 156

Independent claims 116, 128, 137, 146, and 155 each have elements substantially similar to the following form:

> receiving at least one SQL statement, wherein the at least one SQL statement is configured to operate on a first table and a second table;
> automatically generating a set of SQL statements to query the first table and the second table, wherein
> the set of SQL statements are based, at least in part, upon the at least one SQL statement,
> the automatically generating uses a relationship between the first table and the second table to generate the set of SQL statements, and
> the set of SQL statements comprises SQL statements other than the at least one SQL statement;
> producing a first result set by querying the first table using the set of SQL statements;
> producing a second result set by querying the second table using the set of SQL statements, wherein the querying the first table and the querying the second table are performed without joining the first table and the second table;
> joining the first result set and the second result set to produce a third result set; and
> returning the third result set.
> *See, e.g.*, Claim 116.

Appellants respectfully submit that the cited passages of the cited combination of Koo and DiDomizio fail to disclose, teach, or suggest these elements.

While not conceding that the cited references qualify as prior art, but instead to expedite prosecution, Appellants have chosen to respectfully disagree, and demonstrate that the rejections of the FOA are in clear error. Appellants reserve the right, for example, in a continuing application, to establish that the cited references, or other references cited now or hereafter, do not qualify as prior art as to an invention embodiment previously, currently, or subsequently claimed.

(1) Koo and DiDomizio Fail to Teach or Suggest Automatically Generating a Set of SQL Statements Based Upon a Received SQL Statement to Query the First Table and the Second Table

Appellants respectfully submit that Koo and DiDomizio, alone or in any rational combination, fail to teach or suggest all the elements of claim 116, including automatically generating a set of SQL statements based upon a received SQL statement to query the first table and the second table. First, among other deficiencies of Koo, Appellants submit that Koo does not teach or suggest the element of:

> automatically generating a set of SQL statements to query the first table and the second table, wherein
> the set of SQL statements are based, at least in part, upon the at least one SQL statement,
> the automatically generating uses a relationship between the first table and the second table to generate the set of SQL statements, and
> the set of SQL statements comprises SQL statements other than the at least one SQL statement;
> SQL statements other than the at least one SQL statement

Koo is directed to optimizing a database query by analyzing the query to identify any joins within the query that are "lossless" and any tables of the identified joins that are eligible for removal. *See* Koo, Abstract. Koo discusses rewriting an original query to eliminate one or more tables used in that original query. *See* Koo, Abstract. Koo identifies tables that are eligible for removal by characterizing the type(s) and characteristic(s) of joins present in the original query. *See* Koo, 5:26-34. The rewritten query then can be used to access only the tables that are remaining. *Id.*

Appellants submit hat Koo does not teach or suggest "automatically generating a set of SQL statements to query the first table and the second table, wherein the set of SQL statements are based, at least in part, upon the at least one SQL statement." For example, for two exemplary database tables, Koo's query optimization does not automatically generate a set of SQL statements to query the first table and the second table, as claimed. Instead, Koo is directed to optimizing a query by identifying one or more table(s) that can be removed prior to performing the query, and then rewriting the query into a simpler form. In other words, Koo discusses eliminating portions of a given query in order to produce an optimized query. In accordance with the example above, Koo discusses removing one of the two tables and then generating a query to query only the remaining table. Thus, the single query generated by Koo only queries the first table, whereas claim 116 recites automatically generating a set of SQL statements that would then be used to query the first table and the second table.

The FOA, on page 3, agrees with Appellants in this regard, and thus does not cite Koo for the claim element of "automatically generating a set of SQL statements to query the first table and the second table, wherein the set of SQL statements are based, at least in part upon the at least one SQL statement." The FOA cites DiDomizio for this element, citing column 6, line 60 – column 7, line 6 for this element. However, the FOA does cite Koo for a related element of "wherein… the automatically generating uses a relationship between the first table and the second table to generate the set of SQL statements." Appellants find this particular citation peculiar, since as argued above, Koo only discusses removing one of the tables from a query that is being generated. Therefore, any relationship between first and second tables can only be used by Koo when one of the tables is removed from a query being generated.

Therefore, a combination of Koo with any other reference (including DiDomizio) can logically only teach the elimination of either the first table or the second table from the query that is being generated, and thus any such combination cannot teach generating a set of SQL statements that accesses both first and second tables. Any other use of Koo would render the system of Koo inoperable. Put another way, any rational combination of Koo (which discusses using relationship(s) between a first table and a second table to remove one of the tables from a query that is to be performed) with any other reference, that discusses querying two tables (such as DiDomizio), can only teach eliminating either

the first table or the second table from the query that is to be performed, at best. A modification of Koo that might somehow render Koo capable of generating a set of SQL statements that could be used to query both first and second tables, even if such were possible, would render Koo unsatisfactory for its intended purpose. Per MPEP 2143.01 (V), a modification of the cited art cannot render the cited art unsatisfactory for its intended purpose. Therefore, this modification being proposed by the FOA cannot be done, i.e., "there is no suggestion or motivation to make the proposed modification." *See* MPEP 2143.01 (V), citing *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). Koo, therefore, does not teach or suggest at least these features of claim 116.

DiDomizio does not remedy the deficiencies of Koo. DiDomizio is directed to a system that allows a user to access multiple databases, where each database may have a different structure. *See* DiDomizio, Abstract. In DiDomizio's system, a user is presented with an interface for the user to enter unstructured, English-language keyword search. *See* DiDomizio, 4:7-29. DiDomizio's system then generalizes the unstructured search to query a plurality of databases with potentially matching tables. *See* DiDomizio, 4:30-62. DiDomizio's system then presents the user with results from the initial database queries, and the user selects the relevant components from which the final, results-producing database query will be created. *See* DiDomizio, 4:56-61, 9:22-26, and Figs. 5 and 6, among others.

However, DiDomizio also does not teach or suggest "automatically generating a set of SQL statements to query the first table and the second table, wherein the set of SQL statements are based, at least in part upon the at least one SQL statement." Instead, DiDomizio discusses creation of a database query that is accomplished by user-directed selections of the components from which to create the database query. In other words, DiDomizio discusses a manual process of generating a query, which is contrary to the automatic generation of the set of SQL statements of claim 116. Therefore, Appellants respectfully submit that the FOA erroneously cites DiDomizio for disclosing this automatic generation element of claim 116.

Still further, a combination of Koo and DiDomizio does not teach or suggest "automatically generating a set of SQL statements to query the first table and the second table, wherein the set of SQL statements are based, at least in part upon the at least one SQL statement," where the automatically generating uses a relationship between the first

table and the second table to generate the set of SQL statements. Thus, alone or in combination with Koo, DiDomizio does not teach or suggest at least these features of claim 116.

(2) Koo and DiDomizio Fail to Teach or Suggest Producing a First Result Set and a Second Result Set by Querying a First and Second Table, Respectively

Appellants respectfully submit that Koo and DiDomizio, alone or in any rational combination, fail to teach or suggest all the elements of claim 116, including producing a first result set and a second result set by querying a first and second table, respectively. First, Appellants submit that Koo does not teach or suggest "producing a first result set by querying the first table using the set of SQL statements" and "producing a second result set by querying the second table using the set of SQL statements," where "querying the first table and the querying the second table are performed without joining the first table and the second table."

Instead, Koo is directed to rewriting a query to eliminate at least one of tables that are used in an original query. An example of Koo's method performs a rewrite of the following join query:

```
SELECT COUNT(*)
FROM STARS ACCOUNT A, STARS.CUSTOMER C
WHERE A.CUSTID = C.CUSTID AND A.BALANCE < 10
```
Koo 5:5-10.

Koo's example join query operates on two tables, a CUSTOMER table, and an ACCOUNT table. Given the above join query, Koo performs an analysis to determine that the CUSTOMER table can be eliminated, given the join conditions. *See*, Koo 5:11-22. After eliminating the CUSTOMER table, Koo rewrites the above query as the following:

```
SELECT COUNT(*)
FROM STARS ACCOUNT A
WHERE A.BALANCE < 10
```
Koo 5:22-25.

In other words, Koo is directed to deleting at least one table and also rewriting the original query to a simpler query that does not use the deleted table(s). In the example above, Koo discusses deleting the CUSTOMER table before either the original query or the rewritten query is performed. Therefore, Koo cannot teach at least the element of "producing a second result set by querying the second table using the set of SQL statements," since the second table is deleted before any query is performed on that second table (that, in fact, being the point of Koo's approach). Koo, therefore, does not teach or suggest at least these features of claim 116.

DiDomizio does not remedy the deficiencies of Koo. As mentioned above, DiDomizio is directed to create a database query by allowing a user to select components from which to create the database query. However, Appellants do not find any teaching or suggestions in DiDomizio that teach or suggest "producing a first result set by querying the first table using the set of SQL statements" and "producing a second result set by querying the second table using the set of SQL statements," where "querying the first table and the querying the second table are performed without joining the first table and the second table." The FOA, correctly, does not cite DiDomizio for this element of claim 116. Thus, alone or in combination with Koo, DiDomizio does not teach or suggest at least these features of claim 116.

(3) Koo and DiDomizio Fail to Teach or Suggest Joining the First and Second Result Sets to Produce a Third Result Set

Appellants respectfully submit that Koo and DiDomizio, alone or in any rational combination, also fail to teach or suggest all the elements of claim 116, including joining the first and second result sets to produce a third result set. First, Appellants submit that Koo does not teach or suggest "joining, using the processor, the first result set and the second result set to produce a third result set." On page 3, the FOA cites a portion of Koo that is directed to a composite RI join (also referred to a RI join predicate), where "each A row matches one B row, whereas each B row may match multiple A rows." *See* Koo at 6:39-43. In this particular example Koo describes a query with a predicate of "A.FK1=B.PK1 and A.FK2=B.PK2." *See* Koo at 6:38.

However, in this example Koo is merely describing how to "form the RI joins between the parent and child tables." *See* Koo at 6:50-52. In other words, Koo appears

to be describing how to perform a join of two tables, i.e., tables A and B. In contrast, clam 116 recites "joining, using the processor, the first result set and the second result set to produce a third result set," where the first result set is produced "by querying the first table using the set of SQL statements," where the second result set is produced by "querying the second table using the set of SQL statements," and where the "querying the first table and querying the second table are performed without joining the first table and the second table." Appellants respectfully submit that in the cited portions Koo clearly discusses joining the two tables A and B. In contrast, claim 116 distinctly recites that "querying the first table and querying the second table are performed without joining the first table and the second table." Therefore, Koo cannot possibly teach or suggest at least these features of claim 116.

Appellants submit that DiDomizio does not teach or suggest this element. Furthermore, DiDomizio, correctly, is not cited by the FOA for this element either. Thus, alone or in combination with Koo, DiDomizio does not teach or suggest at least these features of claim 116.

Since the combination of Koo and DiDomizio does not teach or suggest each and every feature of claim 116, the combination of Koo and DiDomizio cannot render claim 116 obvious. Furthermore, independent claims 128, 137, 146, and 155 are patentable over Koo and DiDomizio for similar reasons to independent claim 116, and further in view of their own features. Still further, claims dependent on independent claims 116, 128, 137, 146, and 155 are patentable over Koo and DiDomizio for at least the reasons provided for their respective base independent claims, and further in view of their own features. Accordingly, Appellants respectfully request that the rejection of claims 116, 117, 128, 129, 137, 138, 146, 147, 155, and 156 under 35 U.S.C. § 103(a) be reconsidered and withdrawn.

B. Claims 118-125, 130-134, 139-143, 148-152, and 157-161

Appellants respectfully submit that the combination of Koo and DiDomizio fail to disclose, teach, or suggest elements of claim 122. Specifically, regarding claim 122, the FOA alleges that Koo teaches using the first result set in constructing a second set of SQL statements to query the second table, where the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to

another table. However, the cited sections of Koo only disclose the following:

> It is well known in research literature that the parent table (STARS.CUSTOMER in this example) can be eliminated from the query if its columns are not selected, and hence the above query can be rewritten as follows:
>
> SELECT COUNT(*)
> FROM STARS ACCOUNT A
> WHERE A.BALANCE<10
>
> The present invention describes a method of optimally identifying which joins are lossless and which tables are eligible for removal. In this process, the joins are partitioned into lossless or lossy. The parent table of a lossless join is eligible for removal from the query.
> *See* Koo at 5:20-30.

In other words, the cited portions of Koo merely discuss how a parent table can be eliminated from a query. Specifically, this cited portion of Koo is directed to discussing how the CUSTOMER table is eliminated from the following original query:

> SELECT COUNT(*)
> FROM STARS ACCOUNT A, STARS.CUSTOMER C
> WHERE A.CUSTID=C.CUSTID AND A.BALANCE<10
> *See* Koo at 5:6-18.

However, as can be easily discerned from the above sections, Koo is merely discussing how the CUSTOMER table (i.e., parent STARS.CUSTOMER C table) is eliminated from the original query. In other words, Koo discusses deleting at least one table and also rewriting the original query to a simpler query that does not use the deleted table(s). In the example above, Koo discusses deleting the CUSTOMER table before either the original query or the rewritten query is performed.

Even assuming, *arguendo*, that the original query of Koo might somehow be characterized as corresponding to the first set of SQL statements (a view to which Appellants do not acquiesce), and that the re-written query of Koo might somehow be characterized as corresponding to the second set of SQL statements (a view to which Appellants also do not acquiesce), the original query of Koo is never executed (since it is re-written to the re-written form that eliminates the second table). Since the original query of Koo is never executed, the original query of Koo cannot produce a result. Koo provides no teaching or suggestion that such a result is desired or generated. Koo also does not provide any teaching or suggestion how any such result (if it existed) could possibly be used. Furthermore, the FOA also does not provide any explanation or

suggestion on using such a result of Koo, much less equating such result to the first result set of claim 116. Appellants agree with the FOA not making such assertions, since it would not be proper or possible, due to the apparent shortcomings of Koo.

Therefore, Appellants do not see how the FOA can properly allege that in the cited portions of Koo, or anywhere else in Koo, there is any teaching or suggestion of using the first result set in constructing a second set of SQL statements to query the second table, where the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

Appellants also submit that DiDomizio does not remedy the deficiencies of Koo. DiDomizio is (correctly) not cited by the FOA for these elements of claim 116. Thus, alone or in combination with Koo, DiDomizio does not teach or suggest at least these features of claim 116.

Since the combination of Koo and DiDomizio does not teach or suggest each and every feature of claim 122, the combination of Koo and DiDomizio cannot render claim 122 obvious. Furthermore, claims 118-121, 123-125, 130-134, 139-143, 148-152, and 157-161 are patentable over Koo and DiDomizio for similar reasons to claim 122, and further in view of their own features. Accordingly, Appellants respectfully request that the rejection of claims 118-125, 130-134, 139-143, 148-152, and 157-161 under 35 U.S.C. § 103(a) be reconsidered and withdrawn.

C. Claims 126, 127, 135, 136, 144, 145, 153, 154, 162, and 163

Appellants respectfully submit that the combination of Koo and DiDomizio fail to disclose, teach, or suggest elements of claim 126. Specifically, regarding claim 126, the FOA, on page 7, alleges that Koo teaches obtaining a search specification for the query of the first and second tables, wherein the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification. However, the cited sections of Koo only disclose the following:

> Example 1.3.1:
> Columns {FK1,FK2} of table A is a composite foreign key that references table B whose primary key is {PK1,PK2}. Consider the following query predicate:
> AFK1=B.PK1

> Since A.FK2=B.PK2 is absent from the query, the join is N:N, i.e., a lossy join.
>
> Example 1.3.2:
> Column A.FK in table A is a foreign key that references B.PK in table B. Consider the following query predicates: A.FK=B.PK and A.C1=B.C1 If A.C1=B.C1 is avoidable (i.e., EXTRA/DROPPED), then the join is a 1:N join. If A.C1=B.C1 is unavoidable (i.e., CLEAN), then the join is a N:N join.
> *See* Koo at 7:20-35.

In other words, in the cited sections, Koo merely discusses determining whether query predicates for certain tables are directed to lossy joins, 1:N joins, or to N:N joins. Nowhere in the cited portions does Koo discuss any element resembling a search specification. The FOA also does not provide any such mapping and/or explanation of which element of the cited portion of Koo could correspond to the "search specification" of claim 126. Applicants respectfully submit that this is the case, because such a correspondence does not, in fact, exist. Furthermore, nowhere in the cited portions does Koo discuss any element that might somehow successfully be characterized as even resembling the claimed set of SQL statements that comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification. The FOA also does not provide any such mapping and/or explanation of which element(s) of the cited portion of Koo correspond to the above-mentioned element of claim 126.

Appellants also submit that DiDomizio does not remedy the deficiencies of Koo. DiDomizio is (correctly) not cited by the FOA for these elements of claim 126. Thus, alone or in combination with Koo, DiDomizio does not teach or suggest at least these features of claim 126.

Since the combination of Koo and DiDomizio does not teach or suggest each and every feature of claim 126, the combination of Koo and DiDomizio cannot render claim 126 obvious. Furthermore, claims 127, 135, 136, 144, 145, 153, 154, 162, and 163 are patentable over Koo and DiDomizio for similar reasons to claim 126, and further in view of their own features. Accordingly, Appellants respectfully request that the rejection of claims 126, 127, 135, 136, 144, 145, 153, 154, 162, and 163 under 35 U.S.C. § 103(a) be reconsidered and withdrawn.

CONCLUSION

For the above reasons, Appellants respectfully submit that the rejection of pending Claims 116-163 is unfounded. Accordingly, Appellants respectfully request that the Board reverse the rejections of these claims.

If any extensions of time under 37 C.F.R. § 1.136(a) are required in order for this submission to be considered timely, Appellants hereby petition for such extensions. Appellants also hereby authorize that any fees due for such extensions or any other fee associated with this submission, as specified in 37 C.F.R. § 1.16 or § 1.17, be charged to deposit account 502306.

Respectfully submitted,

/ Samuel G. Campbell III /

Samuel G. Campbell III
Attorney for Appellants
Reg. No. 42,381
CAMPBELL STEPHENSON LLP
11401 Century Oaks Terrace
Building H, Suite 250
Austin, Texas 78758
Telephone: (512) 439-5084
Facsimile: (512) 439-5099

CLAIM APPENDIX

Listing of Claims

1 - 115. Canceled.

116. A method comprising:

receiving at least one SQL statement, at a computer system, wherein

the at least one SQL statement is configured to operate on a first table and a second table, and

the at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table;

automatically generating, using a processor of the computer system, a set of SQL statements to query the first table and the second table, wherein

the set of SQL statements are based, at least in part, upon the at least one SQL statement,

the first table and the second table are stored in a computer-readable storage medium of the computer system,

the automatically generating uses a relationship between the first table and the second table to generate the set of SQL statements, and

the set of SQL statements comprises SQL statements other than the at least one SQL statement;

producing a first result set by querying the first table using the set of SQL statements, wherein

the querying the first table is performed using the processor;

producing a second result set by querying the second table using the set of SQL statements, wherein

the querying the second table is performed using the processor, and

the querying the first table and the querying the second table are performed without joining the first table and the second table;

joining, using the processor, the first result set and the second result set to produce a third result set; and

returning the third result set, in response to the receiving the at least one SQL statement.

117. The method of claim 116 wherein the relationship comprises:

a parent/child relationship between the first and second tables, wherein

one of the first and second tables is a parent table, and

if the first table is the parent table, the second table is a child table, and

if the second table is the parent table, the first table is the child table.

118. The method of claim 117 further comprising:

querying the parent table using the set of SQL statements to produce the first result set; and

using the first result set in constructing a second set of SQL statements to query the child table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

119. The method of claim 118 further comprising:

querying the child table using the second set of SQL statements to produce the second result set.

120. The method of claim 119 wherein

the third result set depends on the querying the first table and the querying the second table.

121. The method of claim 118 wherein

the second set of SQL statements comprises:

a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the first result set.

122. The method of claim 116 further comprising:

using the first result set in constructing a second set of SQL statements to query the second table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

123. The method of claim 122 further comprising:

querying the second table using the second set of SQL statements to produce the second result set.

124. The method of claim 123 further comprising:

returning the third result set as a result of the query of the first and second tables.

125. The method of claim 122 wherein the second set of SQL statements comprises:

a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the first result set.

126. The method of claim 116 further comprising:

obtaining a search specification for the query of the first and second tables, wherein

the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification.

127. The method of claim 126 further comprising:

executing the set of SQL statements to produce the third result set; and

returning the third result set in response to the search specification.

128. A system comprising:

a processor;

a memory unit coupled to the processor;

receiving means for receiving at least one SQL statement, wherein

the at least one SQL statement is configured to operate on a first table and a second table, and

the at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table;

generating means for automatically generating a set of SQL statements to query the first table and the second table, wherein

the set of SQL statements are based, at least in part, upon the at least one SQL statement,

the generating means uses a relationship between the first table and the second table to generate the set of SQL statements, and

the set of SQL statements comprise SQL statements other than the at least one SQL statement;

determining means for determining if a parent/child relationship exists between the first and second tables;

first producing means for producing a first result set by querying the first table using the set of SQL statements;

second producing means for producing a second result set by querying the second table using the set of SQL statements, wherein

the querying the first table and the querying the second table are performed without joining the first table and the second table;

joining means for joining the first result set and the second result set to produce a third result set, wherein

the generating means, the determining means, the first querying means, the second querying means and the joining means reside in the memory unit; and

returning means for returning the third result set, in response to receiving the at least one SQL statement.

129. The system of claim 128 further comprising:

parent table determining means for determining if one of the first and second tables is a table, if the parent/child relationship exists, and configured to indicate

if the first table is the parent table, that the second table is a child table, and

if the second table is the parent table, that the first table is the child table, wherein

the parent table resides in the memory unit.

130. The system of claim 129 further comprising:

querying means for querying the parent table using the set of SQL statements to produce the first result set; and

using means for using the first result set in constructing a second set of SQL statements to query the child table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table, and

the querying means and the using means reside in the memory unit.

131. The system of claim 130 wherein

the second querying means is configured to query the child table using the second set of SQL statements to produce the second result set.

132. The system of claim 131 wherein

the result depends on the querying the first table and the querying the second table.

133. The system of claim 130 wherein

the second set of SQL statements comprises:

a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the first result set.

134. The system of claim 128 further comprising:

using means for using the first result set in constructing a second set of SQL statements to query the second table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table, and

said using means resides in the memory unit.

135. The system of claim 128 further comprising:

obtaining means for obtaining a search specification for the query of the first and second tables, wherein

the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification, and

said obtaining means resides in the memory unit.

136. The system of claim 135 further comprising:

executing means for executing the set of SQL statements to produce the third result set; and

returning means for returning the third result set in response to the search specification, wherein

said the executing means and the returning means reside in the memory unit.

137. A computer program product comprising:

receiving instructions to receive at least one SQL statement, wherein

the at least one SQL statement is configured to operate on a first table and a second table, and

the at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table;

generating instructions to automatically generate a set of SQL statements to query the first table and the second table, wherein

the set of SQL statements are based, at least in part, upon the at least one SQL statement,

the generating instructions are configured to use a relationship between the first table and the second table, and

the set of SQL statements comprises SQL statements other than the at least one SQL statement;

first producing instructions to produce a first result set by querying the first table using the set of SQL statements;

second producing instructions to produce a second result set by querying the second table using the set of SQL statements, wherein

the querying the first table and the querying second table are performed without joining the first table and the second table;

joining instructions to join the first result set and the second result set to produce a third result set;

returning instructions to return the third result set, in response to receiving the at least one SQL statement; and

a computer-readable storage medium, wherein

the computer program product is encoded in the computer-readable storage media.

138. The computer program product of claim 137 wherein the relationship comprises:

a parent/child relationship between the first and second tables, wherein

one of the first and second tables is a parent table, if the first table is the parent table, the second table is a child table, and

if the second table is the parent table, the first table is the child table.

139. The computer program product of claim 138 further comprising:

querying instructions configured to query the parent table using the set of SQL statements to produce the first result set; and

using instructions configured to use the first result set in constructing a second set of SQL statements to query the child table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

140. The computer program product of claim 139 wherein
the querying the second table queries the child table using the second set of SQL statements to produce the second result set.

141. The computer program product of claim 140 wherein
the third result set depends on the querying the first table and the querying the second table.

142. The computer program product of claim 139 wherein
the second set of SQL statements comprises:
a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the first result set.

143. The computer program product of claim 137 further comprising:
using instructions configured to use the first result set to construct a second set of SQL statements to query the second table, wherein
the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

144. The computer program product of claim 137 further comprising:
obtaining instructions configured to obtain a search specification for the query of the first and second tables, wherein
the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification.

145. The computer program product of claim 144 further comprising:
executing instructions configured to execute the set of SQL statements to produce the third result set; and

returning instructions configured to return the third result set in response to the search specification.

146. A computer system comprising:

a processor to execute instructions; and

a memory to store the instructions, wherein

the memory is coupled to the processor, and

the instructions comprise:

receiving instructions configured to receive at least one SQL statement, at a computer system, wherein

the at least one SQL statement is configured to operate on a first table and a second table, and

the at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table;

generating instructions configured to automatically generate a set of SQL statements to query the first table and the second table, wherein

the set of SQL statements are based, at least in part, upon the at least one SQL statement,

the generating instructions use a relationship between the first table and the second table to generate the set of SQL statements, and

the set of SQL statements comprises SQL statements other than the at least one SQL statement,

first producing instructions to produce a first result set by querying the first table using the set of SQL statements;

second producing instructions to produce a second result set by querying the second table using the set of SQL statements, wherein

the querying instructions to the first table and the querying instructions to the second table are performed

without joining the first table and the second table;

joining instructions to join the first result set and the second result set to produce a third result set; and

returning instructions to return the third result set, in response to receiving the at least one SQL statement.

147. The computer system of claim 146 wherein the relationship comprises:

a parent/child relationship between the first and second tables, wherein

one of the first and second tables is a parent table,

if the first table is the parent table, the second table is a child table, and

if the second table is the parent table, the first table is the child table.

148. The computer system of claim 147 wherein the instructions further comprise:

querying instructions configured to query the parent table using the set of SQL statements to produce the first result set; and

using instructions configured to use the first result set in constructing a second set of SQL statements to query the child table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

149. The computer system of claim 148 wherein

the querying the second table queries the child table using the second set of SQL statements to produce the second result set.

150. The computer system of claim 149 wherein

the third result set depends on the querying the first table and the querying the second table.

151. The computer system of claim 148 wherein

the second set of SQL statements comprises:

a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the

first result set.

152. The computer system of claim 146 wherein
the instructions further comprise:
using instructions configured to use the first result set to construct a second set of SQL statements to query the second table, wherein the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table.

153. The computer system of claim 146 wherein the instructions further comprise:
obtaining instructions configured to obtain a search specification for the querying of the first and second tables, wherein
the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification.

154. The computer system of claim 154 wherein the instructions further comprise:
executing instructions configured to execute the set of SQL statements to produce the third result set; and
returning instructions configured to return the third result set in response to the search specification.

155. A computer system comprising:
a processor;
a memory unit coupled to the processor;
a receiving module configured to receive at least one SQL statement, wherein
the at least one SQL statement is configured to operate on a first table and a second table, and
the at least one SQL statement comprises an SQL statement that is configured to join the first table and the second table;

a generating module configured to automatically generate a set of SQL statements to query the first table and the second table, wherein

the set of SQL statements are based, at least in part, upon the at least one SQL statement,

the generating module uses a relationship between the first table and the second table, and

the set of SQL statements comprises SQL statements other than the at least one SQL statement;

a first producing module configured to produce a first result set by querying the first table using the set of SQL statements;

a second producing module configured to produce a second result set by querying the second table using the set of SQL statements, wherein

the querying of the first table and the querying of the second table are performed without joining the first table and the second table;

a joining module configured to join the first result set and the second result set to produce a third result set, wherein

the generating module, the determining module, the first producing module, the second producing module and the joining module reside in the memory unit; and

a return output data module configured to return the third result set, in response to receiving the at least one SQL statement.

156. The computer system of claim 155 wherein the relationship comprises:

a parent/child relationship between the first and second tables, wherein

one of the first and second tables is a parent table,

if the first table is the parent table, the second table is a child table,

if the second table is the parent table, the first table is the child table, and

the parent table resides in the memory unit.

157. The computer system of claim 156 further comprising:

a querying module configured to query the parent table using the set of SQL statements to produce the first result set; and

a using module configured to use the first result set in constructing a second set of SQL statements to query the child table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table, and

the querying module and the using module reside in the memory unit.

158. The computer system of claim 157 wherein

the querying the second table queries the child table using the second set of SQL statements to produce the second result set.

159. The computer system of claim 158 wherein

the third result set depends on the querying the first table and the querying the second table.

160. The computer system of claim 157 wherein

the second set of SQL statements comprises:

a query statement for selecting a record having a value of a foreign key field of the second table equal to a value of a target key field in the first result set.

161. The computer system of claim 155 further comprising:

a using module configured to use the first result set to construct a second set of SQL statements to query the second table, wherein

the second set of SQL statements comprises SQL statements other than a second statement that joins the second table to another table, and

said using module resides in the memory unit.

162. The computer system of claim 155 further comprising:

an obtaining module configured to obtain a search specification for the query of the first and second tables, wherein

the set of SQL statements comprises a query statement to select a record from at least one of the first and second tables if the record satisfies the search specification, and

said obtaining module resides in the memory unit.

163. The computer system of claim 162 further comprising:

an executing module configured to execute the set of SQL statements to produce the third result set; and

a returning module configured to return the third result set in response to the search specification, wherein

said the executing module and the returning module reside in the memory unit.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None